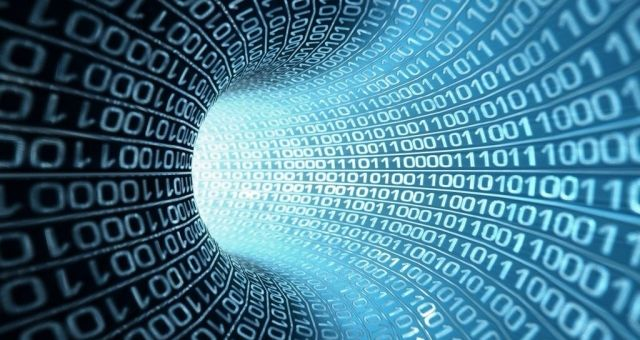
**Walkthrough: Hidden Layers CTF Challenge**

**Challenge Overview**

**File Provided**: 

**Objective**: Extract and decrypt the hidden flag. --------------------------------------------------------------------------------------------------------------------------------------

**Challenge Description**

The given image hides an encrypted flag using Least Significant Bit (LSB) steganography. Your

task is to:

1. Extract the hidden binary data from the LSBs of the blue pixel channel.

2. Convert the binary data into encrypted flag bytes.

3. Use AES decryption to recover the original flag.

--------------------------------------------------------------------------------------------------------------------------------------

**Step 1: Extract Hidden Data Using LSB Steganography**

To extract the data, read the blue channel of the image and extract the least significant bit of each pixel.

from PIL import Image

import numpy as np

image\_path = "hidden\_layers\_challenge.png"

img = Image.open(image\_path).convert("RGB")

pixels = np.array(img)

height, width, \_ = pixels.shape

blue\_channel = pixels[:, :, 2].flatten()

binary\_data = ''.join(str(pixel & 1) for pixel in blue\_channel)

extracted\_bytes = bytes(int(binary\_data[i:i+8], 2) for i in range(0, len(binary\_data), 8))

with open("extracted\_flag.enc", "wb") as f:

f.write(extracted\_bytes)

print("Encrypted flag extracted and saved as extracted\_flag.enc") --------------------------------------------------------------------------------------------------------------------------------------

**Step 2: Decrypt the Extracted Flag**

The flag was encrypted using AES-CBC mode, so you need the AES key and IV. If provided, use them to decrypt the extracted flag.

from Crypto.Cipher import AES

from Crypto.Util.Padding import unpad

# Load extracted encrypted flag

with open("extracted\_flag.enc", "rb") as f:

encrypted\_flag = f.read()

# Provide AES key and IV (from challenge hints or previous extraction)

aes\_key = b'YOUR\_AES\_KEY\_HERE' # Replace with actual key

iv = b'YOUR\_IV\_HERE' # Replace with actual IV

# Decrypt the flag

cipher = AES.new(aes\_key, AES.MODE\_CBC, iv)

decrypted\_flag = unpad(cipher.decrypt(encrypted\_flag), AES.block\_size).decode()

print("Decrypted Flag:", decrypted\_flag)

--------------------------------------------------------------------------------------------------------------------------------------

**Final Flag**

After running the decryption script, the flag should be revealed in the format:

Decrypted Flag: CTF{H1dd3n\_Lay3rs}

--------------------------------------------------------------------------------------------------------------------------------------